A Computational Model of Analogical Reasoning in Dementia Care

Konstantinos Zachos and Neil Maiden
Centre for Creativity in Professional Practice
City University London
Northampton Square, London EC1V 0HB, UK
{k.zachos, N.A.M.Maiden}@city.ac.uk

Abstract

This paper reports a practical application of a computational model of analogical reasoning to a pressing social problem, which is to improve the care of older people with dementia. Underpinning the support for carers for people with dementia is a computational model of analogical reasoning that retrieves information about cases from analogical problem domains. The model implements structure-mapping theory adapted to match source and target domains expressed in unstructured natural language. The model is implemented as a computational service invoked by a mobile app used by carers during their care shifts.

Dementia Care and Creativity

Dementia is a condition related to ageing. After the age of 65 the proportion of people with dementia doubles for every 5 years of age so that one fifth of people over the age of 85 are affected (Alzheimers Society 2010). This equates to a current total of 750,000 people in the UK with dementia, a figure projected to double by 2051 when it is predicted to affect a third of the population either as a sufferer, relative or carer (Wimo and Prince 2010). Dementia care is often delivered in residential homes. In the UK, for example, two in three of all home residents have some form of dementia (e.g. Wimo and Prince 2010), and delivering the required care to them poses complex and diverse problems carers that new software technologies have the potential to overcome. However, this potential is still to be tapped.

The prevailing paradigm in dementia care is person-centered care. This paradigm seeks an individualized approach that recognizes the uniqueness of each resident and understanding the world from the perspective of the person with dementia (Brooker 2007). It can offer an important role for creative problem solving that produces novel and useful outcomes (Sternberg 1999), i.e. care activities that both recognize a sense of uniqueness and are new to the care of the resident and/or carer. However, there is little explicit use of creative problem solving in dementia care, let alone with the benefits that technology can provide. Therefore, the objective of our research was to enable more creative problem solving in dementia care through new software technologies.

This paper reports two computational services developed to support carers to manage challenging behaviors in person-centered dementia care – a computational analogical matching service that retrieves similar challenging behavior cases in less-constrained domains, and a second service that automatically generates creativity prompts based on the computed analogical mappings. Both are delivered to carers through a mobile software app. The next two sections summarize results from one pre-design study that motivates the role of analogical matching in managing challenging behavior in dementia care then describe the two computational creativity services.

A Pre-Design Study

Creative problem solving is not new to care work. Osborn (1965) reported that creative problem solving courses were introduced in nursing and occupational therapy programs in the 1960s. Le Storti et al. (1999) developed a program that fostered the personal creative development of student nurses, challenging them to use creativity techniques to solve nursing problems. This required a shift in nursing education from task- to role-orientation and established a higher level of nursing practice – a level that treated nurses as creative members of health care teams. There have been calls for creative approaches to be used in the care of people with dementia. Successful creative problem solving was recognized to counteract the negative and stressful effects that are a frequent outcome of caring for people with dementia (Help the Aged, 2007). Several current dementia care learning initiatives can be considered creative in their approaches. These include the adoption of training courses in which care staff are put physically into residents' shoes, and exercises to encourage participants to experience life mentally through the eyes of someone with dementia (Brooker 2007). Caring for people with late stage dementia is recognized to require more creative approaches, and a common theme is the need to deliver care specific to each individual's behavioral patterns and habits.

To discover the types of dementia care problem more amenable to this model of creative problem solving, we observed care work and interviews with carers at one UK residential home revealed different roles for creative problem solving in dementia care. One of these roles was to reduce the instances of challenging behavior in residents. Challenging behavior defined as "*culturally abnormal behavior(s) of such an intensity, frequency or duration that the physical safety of the person or others is likely to be placed in serious jeopardy, or behavior which is likely to seriously limit use of, or result in the person being denied access to, ordinary community facilities*" (Bromley and Emerson 1995). Examples include the refusal of food or medication, and verbal aggression.

Interviews with carers revealed that creative problem solving has the potential to generate possible solutions to reduce instances of challenging behavior. For example, if a resident is uncooperative with carers when taking medication, one means to reduce it might be to have a carer wear a doctor's coat when giving the medication. The means is creative because it can be useful, novel to the resident if not applied to him before, and novel to the care team who have not applied it before. Therefore, with carers in the pilot home, we explored the potential of different creativity techniques to reduce challenging behavior.

During one half-day workshop with 6 carers we explored the effectiveness and potential of different creativity techniques to manage a fictional challenging behavior. During a three-stage process the carers were presented with the fictional resident and challenging behavior, generated ideas to reduce the behavior, then prepared to implement these ideas. They used different creativity techniques, presented to them as practical problem solving techniques, to reduce the fictional challenging behavior. The carers demonstrated the greatest potential and appetite for the other exploratory creativity technique, called *Other Worlds* (Innovation Story 2002). During the workshop, the carers sought to generate ideas to reduce the challenging behavior in four different, less constrained domains - *social life*, *research*, *word of mouth* and *different cultures*. These ideas were then transferred to the care domain to explore their effectiveness in it. *Other Worlds* was judged to be the most effective as well as the most interesting to carers. It created more ideas than any of the other techniques, and two of the ideas from the session were deemed sufficiently useful to implement in the pilot home immediately. Carers singled out the technique because, unlike others, it purposefully transferred knowledge and ideas via similarity-based reasoning from sources outside of the immediate problem spaces – the resident, residential home and dementia care domain.

The Carer App

To implement *Other Worlds* in care work we decided to develop a mobile software app, called *Carer*, which carers can use during their work. In the place of human facilitation, the software retrieves then guides carers to explore other worlds that are retrieved by the app, and in place of face-to-face communication, the software was to support asynchronous communication between carers who would digitally share information about care ideas and practices via the software.

The Carer app accesses a digital repository to retrieve natural language descriptions of cases of good care practice in XML based on the structure of dementia care case studies reported by the Social Care Institute for Excellence (Owen and Meyer 2009) as well as challenging behavior cases in non-care domains such as *teen parenting*, *student mentoring* and *prison life*. Each case has two main parts of up to 150 words of prose each – the situation encountered and the care plan enhancement applied – and is attributed to one class of domain to which the case belongs. The current version of the repository contains 115 case descriptions.

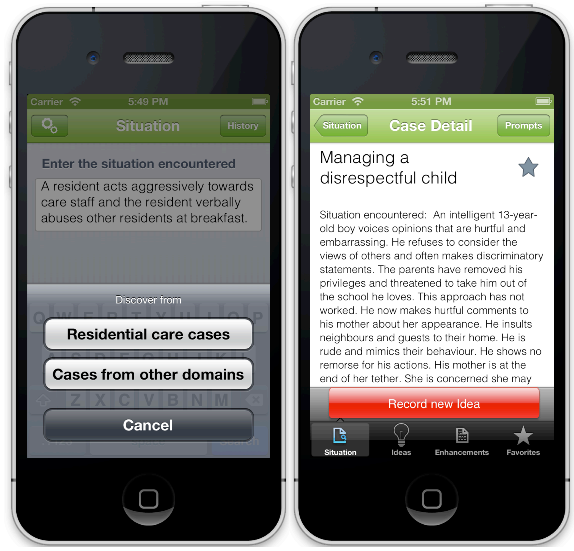

Figure 1. The Carer mobile app showing how carers describe challenging behaviors (on the left-hand side) and a detailed description of one of these cases (on the right-hand side)

Carer app automatically retrieves the previous cases using different services in response to natural language entries typed and/or spoken by a carer into the app. One supports case-based reasoning with literally similar cases based on information retrieval techniques, similar to strategies applied to people with chronic diseases (Houts et al. 1996). A second supports the other worlds technique more generally by automatically generating different domains such as *traveling* or *cooking* in which to generate care plan enhancements to a current situation without the constraints of the care domain (Innovation Company 2002). The user is encouraged to think about how to solve the aggression situation *in the school playground*. A simple flick of the screen will generate a different other world, such as *parachuting from an aircraft*. A third service automatically generates creativity prompts from retrieved case content. Lastly, the *Carer* app invokes AnTiQue, an *analogical reasoning* discovery service that matches the description of a challenging behavior situation to descriptions in the repository of challenging behavior cases in non-care domains. To do this, the service implements a computational analogical reasoning algorithm based on the Structure-

Mapping Theory (Gentner 1983; Falkenhainer et al. 1989) with natural language parsing techniques and a domain-independent verb lexicon called VerbNet (Kipper et al. 2000). A carer can then record new ideas resulting from creative thinking in audio form, then reflect on them by playing them back to change them, generate further ideas, compose them into a care plan and share the plan with other carers. Some of these features are depicted in Figure 1. The right-hand side of Figure 1 shows one retrieved analogical case description – *Managing a disrespectful child* – as it is presented to a carer using the app. The *Carer* app is described at length in Maiden (2012). The next section describes two of the computational creativity services – the analogical reasoning discovery service and the creativity prompt generation service.

The Analogical Reasoning Discovery Service

This service (called AnTiQue) matches a description of challenging behavior in dementia care to descriptions of challenging behavior problems and resolutions in other domains, for example *good policing practices to manage disorderly revelers* and good *teaching practices to manage disruptive children*. AnTiQue's design seeks to solve 2 research problems: (i) match incomplete and ambiguous natural language descriptions of challenging behaviour in dementia care and challenging behaviour problems and resolutions in other domains using different lexical terms; (ii) compute complex analogical matches between descriptions without a priori classification of the described domains.

Analogical service retrieval can increase the number of cases that are useful to the care staff by retrieving descriptions of cases solved successfully in other domains, for example *good policing practices to manage disorderly revelers* and good *teaching practices to manage disruptive children*. The problem and solution description of each case might have aspects that, through analogical reasoning, can trigger discovery of new ideas on the current challenging behaviour. For example, a description of good policing practice to manage disorderly revellers can provide analogical insights with which to manage challenging behaviour in dementia care. AnTiQue seeks to leverage these new sources of knowledge in dementia care.

Analogical retrieval in AnTiQue uses a similarity model called the Structure Mapping Theory (SMT) (Gentner 1983) which seeks to transfer a network of related facts rather than unrelated one (Gentner 1983) from a source to a target domain. To enable structure-matching AnTiQue transforms natural language queries and case descriptions into predicates that express prepositional networks of nodes (objects) and edges (predicate values). Attributional predicates state properties of objects in the form PredicateValue(Object) such as *asleep(resident)* and *absent(relative).* Relational predicates express relations between objects as PredicateValue (Object1,Object2) such as *abuse(resident, care-staff)* and *remain(resident,room)*. According to the SMT, a literal similarity is a comparison in which attributional and relational predicates can both be mapped from a source to a target. In contrast an analogy is a comparison in which relational predicates but few or no attributional predicates can be mapped. Therefore AnTiQue retrieves cases with high match scores for relational predicates and low match scores for attributional predicates, for example a match with the predicate *abuse(detainee,police-officer)* but no match with the predicate *drunk(detainee)*.

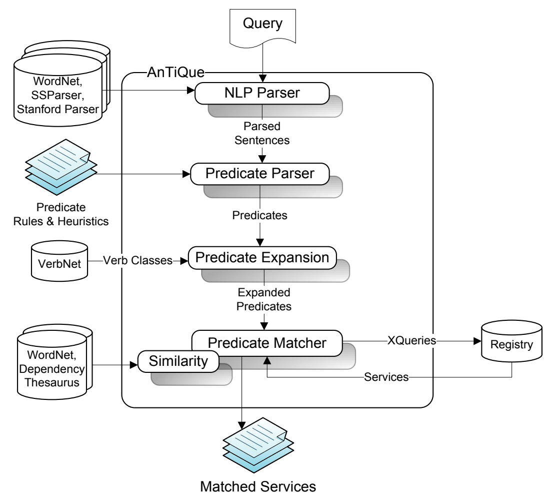

Figure 2. Internal structure of AnTiQue

Figure 2 depicts AnTiQue's 5 components. When invoked the service first divides query and case problem description text into sentences, then part-of-speech tagged, shallow parsed to identify sentence constituents and chunked in noun phrases. It then applies 21 syntax structure rules and 7 lexical extraction heuristics to identify predicates and extract lexical content in each sentence. Natural language sentences are presented as predicates in the form *PredicateValue(Object1, Object2)*. The service then expands each query predicate with additional predicate values that have similar meaning according to verb classes found in VerbNet to increase the likelihood of a match with predicates describing each case. For example the predicate value *abuse* is in the same verb class as *attack*. The service then matches all expanded predicates to a similar set of predicates that describe the problem description of each case in the repository. This is achieved using XQuery text- searching functions to discover an initial set of cases that satisfy global search constraints. Finally it applies semantic and dependency-based similarity measures to refine the candidate case study set. The service returns an ordered set of analogical cases based on the match score with the query.

The components use WordNet, VerbNet, and the Dependency Thesaurus to compute attributional and relational similarities. WordNet is a lexical database inspired by psycholinguistic theories of human lexical memory (Miller

1993). Its word senses and definitions provide the data with which to disambiguate terms in queries and case problem descriptions. Its semantic relations link terms to other terms with similar meanings with which to make service queries more complete. For example a service query with the term *car* is expanded with other terms with similar meaning, such as *automobile* and *vehicle*, to increase matches with web service descriptions.

VerbNet (Kipper et al. 2000) is a domain independent verb lexicon. It organizes terms into verb classes that refine Levin classes (Levin 1993) and add sub-classes to achieve syntactic and semantic coherence among members of a verb class. AnTiQue uses it to expand query predicate values with different members from the same verb class. For example, queries with the verb *abuse* are expanded with other verbs with similar meaning such as *attack*.

The Dependency Thesaurus supports dependency-based word similarity matching to detect similar words from text corpora. Lin (1998) used a 64-million word corpus to compute pair-wise similarities between all of the nouns, verbs, adjectives and adverbs in the corpus using a similarity measure. Given an input word the Dependency Thesaurus can retrieve similar words and group them automatically into clusters. AnTiQue used the Dependency Thesaurus to compute the relational similarity between 2 sets of predicates.

In the remainder of this section we demonstrate the AnTiQue components using text from the following example challenging behaviour situation:

> *A resident acts aggressively towards care staff and the resident verbally abuses other residents at breakfast. Suspect underlying insecurities to new people.*

Natural Language Processing

This component prepares the structured natural language (NL) service query for predicate parsing and expansion. In the first step the text is split into sentences. In the second a part-of-speech tagging process is applied that marks up the words in each sentence as corresponding to a particular lexical category (part-of-speech) using its definition and context. In the third step the algorithm applies a NL processing technique called *shallow parsing* that attempts to provide some machine understanding of the structure of a sentence without parsing it fully into a parsed tree form. The output is a division of the text's sentences into a series of words that, together, constitute a grammatical unit. In our example the tagged sentence *a resident acts aggressively towards care staff and the resident verbally abuses other residents at breakfast* is shown in Figure 3. Tags that follow a word with a forward slash (e.g. *driver/NN*) correspond to lexical categories including noun, verb, adjective and adverb. For example, the *NN* tag means "noun singular or mass", *DT* means "determinant" and *VBZ* means "verb, present tense, 3rd person singular". Tags attached to each chunk (e.g. *[The/DT driver/NN]NP*) correspond to phrasal categories. For instance, the *NP* tag denotes a "noun phrase", *VP* a "verb phrase", *S* a "simple declarative clause", *PP* a "prepositional phrase" and *ADVP* a "adverb phrase".

[A/DT resident/NN]NP [acts/VBZ]VP [aggressively/RB]ADVP [towards/]PP [care staff/NN]NP.

Figure 3. The sentence *a resident acts aggressively towards care staff* after performing part-of-speech tagging and chunking

The component then decomposes each sentence into its *phrasal categories* used in the next component to identify predicates in each sentence structure.

Predicate Parsing

This component automatically identifies predicate structures within each annotated NL sentence based on *syntax structure rules* and *lexical extraction heuristics*. Syntax structure rules break down a pre-processed NL sentence into sequences of phrasal categories where each sequence contains 2 or more phrasal categories. Lexical extraction heuristics are applied on each identified sequence of phrasal categories to extract its lexical content used to generate one or more predicates.

Firstly the algorithm applies 21 syntax structure rules. Each rule consists of a phrasal category sequence of the form *Ri* → *[Bj]*, meaning that the rule *Ri* consists of a phrasal category sequence *B1, B2*,…, *Bj*. For example the rule *R4* → *[NP, VP, S, VP, NP]* reads: rule *R1* consists of a *NP* followed by a *VP*, a *S*, a *VP*, and a *NP*, where *NP*, *VP* and *S* mean a *noun phrase*, a *verb phrase* and a *simple declarative clause* respectively. The method takes a phrasal category list as input and returns a list containing each discovered syntax structure rule and its starting point in the corresponding phrasal category list, e.g. *{(R1,3), (R5,1)}*. In our example, the input for the pre-processed sentence shown in Figure 3 corresponds to a list *Input = (NP, VP, ADVP, PP, NP)*. Starting from the first list position the method recursively checks whether there exists a sequence within the phrasal category list that matches one of the syntax structure rules. The output after applying the algorithm on list *Input* is a list of only one matched syntax structure rule, i.e. *Output = {(R2,1)}*.

Secondly the algorithm applies lexical extraction heuristics on a syntax structure rule-tagged sentence to extract content words for generating one or more predicates. For each identified syntax structure rule in a sentence the algorithm: (1) determines the position of both noun and verb phrases within the phrasal category sequence; (2) applies the heuristics to extract the content words (verbs and nouns) from each phrase category; (3) converts each verb and noun to its morphological root (e.g. *abusing* to *abuse*); and (4) generates the corresponding predicate p in the form *PredicateValue(Object1, Object2)* where *PredicateValue* is the verb and *Object1* and *Object2* the nouns. To illustrate this the algorithm identified rule *R2+* for our example sentence in Figure 3. According to one heuristic *{R2+}* corresponds to the following phrasal category sequence *[NP, VP, ADVP, PP, NP]*. Therefore the algorithm determines the position of both noun and verb phrases within this sequence, i.e. noun phrases in *{NP,1}* and *{NP,5}* and verb phrases in *{VP,2}*. Lexical extraction heuristics are applied to extract the content words from each phrase category, i.e. *{NP,1}* → *resident*, *{NP,5}* → *care staff*, *{VP,2}* → *act*. Returning to our example, the algorithm generates two predicates for the sentence *a resident acts aggressively towards care staff and the resident verbally abuses other residents at breakfast*, namely *act(resident,care_staff)* and *abuse(resident,resident)*.

Predicate Expansion

Predicate expansion and matching are key to the service's effectiveness. In AnTiQue queries are expanded using words with similar meaning. AnTiQue uses ontological information from VerbNet to extract semantically related verbs for verbs in each predicate.

VerbNet classes are organised to ensure syntactic and semantic coherence among members, for example the verb *abuse* as *repeatedly treat a victim in a cruel way* is one of 24 members of the *judgement* class. Other members include *attack, assault* and *insult* and 20 other verbs as potential expansions. Thus VerbNet provides 23 verbs as potential expansions for the verb *abuse*. Although classes group together verbs with similar argument structures, the meanings of the verbs are not necessarily synonymous. For instance, the degree of attributional similarity between *abuse* and *reward* is very low, whereas the similarity between *abuse* and *assault* is very high. The service constrains use of expansion to verb members that achieve a threshold on the degree of attributional similarity computed with WordNet-based similarity measurements (Simpson and Dao 2005). Given 2 sets of text, $T1$ and $T2$, the measurement determines how similar the meaning of $T1$ and $T2$ is scored between 0 and 1. For example, for the verb *abuse*, the algorithm computes the degree of attributional similarity between *abuse* and each co-member within the *judgement* class. In our example verbs such as *attack*, *assault* and *insult* but not *honour* and *doubt* are used to generate additional predicates in the expanded query.

Predicate Matching

Coarse-grained Matching The expanded query is fired at problem descriptions of cases in the repository as an XQuery. Prior to executing the XQuery we pre-process all problem descriptions of cases in the registries using the Natural Language Processing and Predicate Parsing components and store them locally. The XQuery includes functions to match each original and expanded predicate value to equivalent representations of candidate problem descriptions of cases. The service retrieves an initial set of matched cases.

Fine-grained Matching The Predicate Matcher applies semantic and dependency-based similarity measures to assess the quality of the candidate case set. It computes relational similarity between the query and each case retrieved during coarse-grain matching. To compute relational similarities that indicate analogical matches between service and query predicate arguments the Predicate Matcher uses the Dependency Thesaurus to select web services that are relationally similar to mapped predicates in the service query.

In our example the case *Managing a disrespectful child*, which describes a good childcare practice to manage a disrespectful child, is one candidate case retrieved during coarse-grained matching. Figure 4 shows the problem and solution description of the case.

Name	Managing a disrespectful child
Problem	An intelligent 13-year-old boy voices opinions that are hurtful and embarrassing. The child refuses to consider the views of others and often makes discriminatory statements. The parents have removed his privileges and threatened to take him out of the school he loves. This approach has not worked. He now makes hurtful comments to his mother about her appearance. The child insults neighbours and guests at their home. He is rude and mimics their behaviour. The child shows no remorse for his actions. His mother is at the end of her tether.
Solution	The son needs very clear boundaries set. The parents are going to set clear rules on acceptable behaviour. They will state what they are not prepared to tolerate. They will highlight rude comments in a firm tone with the boy. He will receive an explanation as to why the comments are hurtful. Both parents will agree punishments for rule breaking that are realistic. They will work as a team and follow through on punishments. The son can then regain his privileges as rewards for consistent good behaviours.

Figure 4. A retrieved case describing a good childcare practice to manage a disrespectful child

The algorithm receives as inputs a pre-processed sentence list for the query and problem description of the case. It compares each predicate in the pre-processed query sentence list $Pred(j)_{Query}$ with each predicate in the pre-processed problem description sentence list $Pred(k)_{Case}$ to calculate the relevant match value, where

$$Pred(j)_{Query} = PredVal_{Query}(Arg1_{Query};\ Arg2_{Query})$$

and

$$Pred(k)_{Case} = PredVal_{Case}\ (Arg1_{Case};\ Arg2_{Case}).$$

The following conditions must be met in order to accept a match between the predicate pair:

1. $PredVal_{Case}$ exists in list of expanded predicate values of $PredVal_{Query}$;
2. $Arg1_{Query}$ and $Arg1_{Case}$ (or $Arg2_{Query}$ and $Arg2_{Case}$ respectively) are not the same;
3. $Arg1_{Case}$ (or $Arg2_{Case}$) exists in the Dependency Thesaurus result set when using $Arg1_{Query}$ (or $Arg2_{Query}$) as the query to the Thesaurus;
4. the resulting attributional similarity value from step 3 is below a specified threshold.

If all conditions are met, $Pred_{Case}$ is added to the list of matched predicates for the current case. If not the algorithm rejects $Pred_{Case}$ and considers the next list item.

AnTiQue queries the Dependency Thesaurus to retrieve a list of dependent terms. Terms are grouped automatically according to their dependency-based similarity degree. Firstly the algorithm checks whether the case predicate argument exists in this list. If so, it uses the semantic similarity component to further refine and assess the quality of the case predicate with regards to relational similarity.

Using this 2-step process AnTiQue returns an ordered set of analogical cases based on the match score with the query. In our example consider $Pred(j)_{Query}$ = *abuse(resident,residents)* extracted from the sentence *the resident verbally abuses other residents at breakfast,* and the $Pred(k)_{Case}$ = *insult(child,neighbours)* from the sentence *The child insults neighbours and guests at their home* taken from the description of the *Managing a disrespectful child* good childcare practice case in Figure 4. In this example all conditions for an analogical match are met: the predicate values *abuse* and *insult* are semantically equivalent whilst the object names *resident* and *child* and *residents* and *neighbours* are not the same. According to the Dependency thesaurus *child* is similar based on dependencies to *resident, and neighbour* is similar based on dependencies to *resident*. Finally the attributional similarity value of *resident* and *child* is 0.33, for *resident* and *neighbour* 0.25 – both below the specified threshold. As a result the predicate *insult(child,neighbours)* is added to the list of matched predicates for the predicate *abuse(resident,resident)*.

At the end of each invocation, the service returns an ordered set of the descriptions of the highest-scoring cases for the app component to display to the care staff.

The Creativity Trigger Generation Service

Although care staff can generate new resolutions directly from retrieved case descriptions, formative usability testing with the app revealed that users were often overwhelmed by the volume of text describing each case and uncertain how to start idea generation. Therefore we developed an automated service that care staff can invoke to generate creative triggers that extract content from the retrieved descriptions to conjecture new ideas that care staff can consider for the resident. Each trigger expresses a single idea that care staff can use to initiate creative thinking. The service uses the attributional predicates generated by the analogical matching discovery service to generate prompts that encourage analogical transfer of knowledge using the object-pair mappings identified in each predicate. It has the form *Think about a new idea based on the*, followed by mapped subject and object names in the target domain. To illustate, referring back to the *Managing a disrespectful child* good practice case retrieved from the childcare domain shown in Figure 1, Figure 5 shows how they are presented in the Carer mobile app while Figure 6 lists all creativity prompts that the service generates for the analogical case.

Figure 5. The Carer mobile app showing creativity prompts generated for the *Managing a disrespectful child* case

Think about a new idea based on the boundaries
Think about a new idea based on the clear rules
Think about a new idea based on the acceptable behaviour
Think about a new idea based on the rude comments
Think about a new idea based on the firm tone
Think about a new idea based on the explanation
Think about a new idea based on the comments
Think about a new idea based on the punishment
Think about a new idea based on the rule breaking
Think about a new idea based on the rewards
Think about a new idea based on the privileges
Think about new idea based on the consistent good behaviour

Figure 6. Creativity prompts generated for the *Managing a disrespectful child* case

Discovering Novel Ideas

Our design of the Carer app builds on Kerne et al. (2008)'s notion of human-centered creative cognition, in which information gathering and idea discovery occur concurrently, and information search and idea generation reinforce each other. The computational model of analogical reasoning searches for and retrieves information from analogical domains, and the creativity trigger generation service manipulates this information to support more effective idea generation from information, however the generation of new ideas remains a human cognitive activity undertaken by carers, supported by bespoke features implemented in the app.

For example, a carer can audio-record a new idea at any time in response to retrieved analogical cases and/or presented creativity triggers by pressing the red button visible in Figures 1 then verbalizing and naming the idea. Recorded ideas can be selected and ordered to construct a new care enhancement plan that can be extended with more ideas and comments at any time. The carer can also play back the audio-recorded ideas and care enhancement plans to reflect and learn about them, inspired by similar use of the audio channel in digitally supported creative brainstorming (van Dijk et al. 2011). Reflection about an idea is supported with guidance from the app to reflect on why the idea is needed, what the idea achieved, and how and when the idea should be implemented. Reflection about a care enhancement plan is more sophisticated. A carer can drag-and-drop ideas in and out of the plan and into different sequences in it. Then, during play back of the plan, the app concatenates the individual idea audio files and plays the plan as a single recording, allowing the carer to listen to and reflect on each version of the plan as a different narrated story. Moreover, s/he can reflect collaboratively with colleagues using the app to share the plan as e-mail attachments, thereby enabling asynchronous communication between carers.

Formative Evaluation of the Carer App

The *Carer* app was made available for evaluation over prolonged periods with carers in a residential home. At the start of the evaluation, 7 nurses and care staff in the residential home were given an iPod Touch for their individual use during their care work over a continuous 28-day period. All 7 carers received face-to-face training in how to use the device and both apps before the evaluation started. A half-day workshop was held at the residential home to allow them to experiment with all of both apps' features. The carers were also given training and practice with the 3 forms of *Other Worlds* creativity technique through practice and facilitation to demonstrate how it can lead to idea generation. We deemed this training in the creativity technique an essential precondition for successful uptake of the app.

Even though it only lasted 4 weeks, the reported evaluation of the *Carer* app in one residential home provided valuable data about the use of mobile computing and creativity techniques in dementia care. Figure 7 depicts the results.

	Residential cases	Analogical domain cases	Ideas generated	Enhancement plans generated
Totals	27	5	14	10

Figure 7. Situations, ideas and care enhancement plans generated by care staff using Carer app

The focus group revealed that the nurses and carers implemented at least one major change to the care of one resident based on ideas generated using the app.

However, most of this success was not based on the analogical cases retrieved by the computational model. Whilst carers using the app did use the analogical matching service, and the service did retrieve relevant cases from analogical domains such as childcare and student management, the carers were unable to map and transfer knowledge from each of these source domains to the current dementia-related challenging behavior. The log data recorded only 5 uses of the *analogical reasoning* service to retrieve descriptions of cases of challenging behaviors from non-care domains. Rather, the carers appeared to use the *case-based reasoning* service to retrieve descriptions of challenging behavior cases from the care domain – the log data recorded 28 uses of this service, and most of the 114 recorded uses of the *creativity prompt generation* service were generated from these same-domain dementia cases. The focus group revealed that the carers did not use retrieved non-care domain cases because they were unable to recognize analogical similarities between them and the challenging behavior situation. We identified two possible reasons for this. Firstly, AnTiQue implements an approach that *approximates* analogical retrieval, hence there is always the possibility of computing seemingly "wrong" associations and retrieve cases that do not have analogical similarities. Previous evaluations of AnTiQue with regards to the precision and recall (Zachos & Maiden, 2008) revealed a recall score of 100% and a precision score of 66,6% highlighting one potential limitation of computing the attributional similarity using WordNet-based similarity measures.

Secondly, the results suggests that carers will require more interactive support based on results generated by the computational model to support cognitive analogical reasoning, consistent with previously reported empirical findings (e.g. Gick 1983). Examples of such increased interactive support include explicitly reporting each computed analogical mapping to the carer, use of graphical depictions of structured knowledge to transfer from the source to the target domain, and more deliberate analogical support prompts, for example based on the form *A is to B as C is to D*. We are extending Carer app with such features and look forward to reporting these extensions in the near future.

Related Work

Since the 1980s, the efforts of many Artificial Intelligence researchers and psychologists have contributed to an emerging agreement on many issues relating to analogical reasoning. In various ways and with differing emphases, all current computational analogical reasoning techniques use underlying structural information about the sources and the target domains to derive analogies. However, at the algorithmic level, they achieve the computation in many different ways (Keane et al. 1994).

Based on the Structure Mapping Theory (SMT), Gentner constructed a computer model of this theory called Structure Mapping Engine (SME) (Gentner 1989). The method assumes that both target and source situations are represented using a certain symbolic representation. The SME also only uses syntactic structures about the two situations as the main input knowledge — it has no knowledge of any kind of semantic similarity between various descriptions and relations in the two situations. All processing is based on syntactic structural features of the two given representations.

The application of analogical reasoning to software reuse is not new. For example, Massonet and van

Lamsweerde (1997) applied analogy-making techniques to complete partial requirements specifications using a rich, well-structured ontology combined with formal assertions. The method was based on query generalization for completing specifications. The absence of effective ontologies and taxonomies would expose the weaknesses of the proposed approach due to the reliance on ontologies. Pisan (2000) tried to overcome this weakness by applying the SME to expand semi-formal specifications. The idea was to find mappings from specifications for problems similar to the one in hand and use the mappings to adapt an existing specification without requiring domain specific knowledge. The research presented in this paper overcomes limitations of the above-mentioned approaches by using additional knowledge bases to extent the mapping process with semantic similarity measures.

Conclusion and Future Work

This paper reports a practical application of a computational model of analogical reasoning to a pressing social problem, which is to improve the care of older people with dementia. The result is a mobile app that is capable technically of accepting spoken and typed natural language input and retrieving analogical domain cases that can be presented with creativity triggers to support analogical problem solving.

The evaluation results reported revealed that our model of creative problem solving in dementia care did not describe all observed carer behavior, so we are currently repeating the rollout and evaluation of *Carer* in other residential homes to validate this finding. *Carer* is being extended with new creativity support features that include web images that match generated creativity prompts, and more explicit support for analogical reuse of cases from non-dementia care domains. We are extending the repository with new cases that are semantically closer to dementia care and, therefore, easier to recognize analogical similarities with.

Acknowledgment

The research reported in this paper is supported by the EU-funded MIRROR integrated project 257617, 2010-14.

References

Alzheimers Society, 2010. Statistics. http://www.alzheimers.org.uk/site/scripts/documents_info.php?documentID=341

Wimo A. & Prince M., 2010, 'World Alzheimer Report 2010: The Global Economic impact of Dementia, http://www.alz.co.uk/research/worldreport/

Brooker, D., 2007, 'Person-centred dementia care: Making Services Better, Bradford Dementia Group Good Practice Guides. Jessica Kingsley Publishers London and Philadephia

Osborn A., 1965. The Creative Trend in Education. In: Source Book For Creative Problem Solving: A Fifty Year Digest of Proven Innovation Processes. Creative Education Foundation Press, New York.

Le Storti A., J., Cullen P., A., Hanzlik E., M., Michiels, J., M., Piano L., A., Lawless Ryan P., Johnson W., 1999. Creative Thinking In Nursing Education: Preparing for Tomorrow's Challenges. Nursing Outlook. Vol. 47, no. 2 62–66.

Help The Aged, 2007, 'My Home Life: Quality of Life In Care Homes; A Review of the Literature London. [online] http://myhomelifemovement.org/ downloads/mhl_review.pdf [Accessed 5 Jan 2011].

Bromley J. and Emerson E., 1995, 'Beliefs and Emotional Reactions of Care Staff working with People with Challenging Behaviour', Journal of Intellectual Disability Research 39(4), 341-352

Innovation Company, 2002. 'Sticky Wisdom (?WhatIf!)'. Capstone Publishing Company Limited, Chichester.

Owen T. and Meyer J., 2009, 'Minimizing the use of 'restraint' in care homes: Challenges, dilemmas and positive approaches', Adult Services Report 25, Social Care Institute of Excellence, http://www.scie.org.uk/publications/reports/report25.pdf.

Houts, P.S., Nezubd, A.M., Magut Nezubd, C, Bucherc, J.A., 1996, 'The prepared family caregiver: a problem- solving approach to family caregiver education', Patient Education and Counselling, 27,1, 63-73.

Gentner D., 1983, 'Structure-Mapping: A Theoretical Framework for Analogy', Cognitive Science 5, 121-152.

Falkenhainer B., Forbus K.D. & Gentner D., 1989, 'The Structure-Mapping Engine: Algorithm and Examples', Artificial Intelligence 41, 1-63.

Kerne, A., Koh, E., Smith, S. M., Webb, A., Dworaczyk, B., 2008, 'combinFormation: Mixed-Initiative Composition of Image and Text Surrogates Promotes Information Discovery', ACM Transactions on Information Systems, 27(1), 1-45.

Kipper K., Dang H.T. and Palmer M., 2000, 'Class-based Construction of a Verb Lexicon', Proceedings AAAI/IAAI Conference 2000, 691–696.

Miller K., 1993, 'Introduction to WordNet: an On-line Lexical Database' Distributed with WordNet software.

Levin B., 1993, 'English Verb Classes and Alternations: A Preliminary Investigation', University Chicago Press.

Lin D., 1998, 'Automatic retrieval and clustering of similar words', In COLINGACL, 768-774.

Maiden N.A.M., 2012, 'D5.2 Deliverable: Techniques and Software Apps for Integrated Creative Problem Solving and Reflective Learning Version 1', Technical Report. Available at http://www.mirror-project.eu/showroom-a-publications/deliverables.

Simpson, T. and Dao, T. (2005). Wordnet-based semantic similarity measurement. codeproject.com/cs/library/semanticsimilaritywordnet.asp

van Dijk J., van der Roest J., van der Lugt R. & Overbeeke K., 'NOOT: A Tool for Sharing Moments of Reflection during Creative Meetings', Proceedings 10th ACM Creativity and Cognition Conference, Atlanta Georgia, Nov 2011, ACM Press.

Gick M.L., 1989, 'Two Functions of Diagrams in Problem Solving by Analogy', Knowledge Acquisition from Text and Pictures', ed. M. Handi & J.R. Levin, Elsevier Publishers B.V. North-Holland, 215-231.

Keane, M.T., Ledgeway, T. & Duff, S. (1994). Constraints on analogical map- ping: A comparison of three models. Cognitive Science, 18, 387–438.

Massonet, P. and van Lamsweerde, A. (1997). Analogical reuse of requirements framework. 3rd IEEE International Symposium on Requirements Engineering.

Pisan, Y. (2000). Extending requirement specifications using analogy. 22nd International Conference on Software Engineering (ICSE), limerick, Ireland.

Zachos K. & Maiden N.A.M., 2008, 'Inventing Requirements from Software: An Empirical Investigation with Web Services', Proceedings 16th IEEE International Conference on Requirements Engineering, IEEE Computer Society Press, 145-154.